

In the Claims

1(Previously Presented). A control method for an extensible markup language file, comprising the steps of:

- a) receiving a query from a user;
- b) determining an access control rule associated with the user;
- c) performing a query search on the extensible markup language file;
- d) storing a query search result;
- e) performing an access search on the extensible markup language file;
- f) storing an access search result; and
- g) comparing the query search result and the access search result to determine an allowed search result.

2(Original). The method of claim 1, wherein step (g) further includes the steps of:

- g1) comparing a search convergence depth to an access convergence depth;
- g2) when the search convergence depth is equal to the access convergence depth, performing an intersection between the query search result and the access search result to form an intersection result;
- g3) when the access rule is a hide command, the allowed search result is a non-intersecting result.

3(Original). The method of claim 2, further including the step of:

- g4) when the access rule is a show command, the allowed search result is the intersection result.

4(Original). The method of claim 2, further including the steps of:

g4) when the search convergence depth is greater than the access convergence depth, converging the query search result to the access convergence depth to form a converged query search result;

g5) performing an intersection between the convergence search result and the access search result to form the intersection result.

5(Original). The method of claim 2, further including the step of:

g4) when the search convergence depth is less than the access convergence depth, performing an intersection between the query search result and the access search result to form the intersection result.

6(Original). The method of claim 1, wherein step (a) further includes the step of:

a1) converting the query into an execution stack.

7(Original). The method of claim 6, wherein step (a1) further includes the step of:

i) creating a line of the execution stack that contains an operation, a convergence depth, a term and a pattern.

8(Original). The method of claim 6, wherein step (a1) further includes the step of:

i) creating a plurality of lines of the execution stack that contain an operation, a convergence depth, a term and a pattern;

ii) executing the plurality of lines of the execution stack to form a plurality of results;

9(Original). The method of claim 1, wherein step (c) further includes the steps of:

c1) flattening the extensible markup language file to form a flattened extensible markup language file.

10(Original). The method of claim 9, further including the step of:

c2) returning a line number of the flattened extensible markup language file.

11(Original). The method of claim 1, wherein step (g) further includes the step of:

g1) performing an intersection between a plurality of line numbers from the query search result and a second plurality of line numbers from the access search result.

12(Previously Presented). A control method for an extensible markup language file, comprising the steps of:

- a) determining an access control rule for a user;
- b) receiving a query from the user;
- c) performing an access control search against the extensible markup language file to form an access control result;
- d) performing a query search against the extensible markup language file to form a query search result; and
- e) comparing the access control result and the query search result to determine an allowed search result.

13(Original). The method of claim 12, wherein step (a) further includes the step of:

a1) creating an access control query statement.

14(Original). The method of claim 12, wherein step (b) further includes the step of:

b1) converting the query into an executable stack.

15(Original). The method of claim 12, wherein step (a) further includes the step of:

a1) creating an access control query statement containing a predicate.

16(Previously Presented). A control method for an extensible markup language file, comprising the step of:

a) performing an access search on the extensible markup language file to form an access search result;

b) performing a query search on the extensible markup language file to form a query search result; and

c) comparing the access search result to the query search result to form an allowed search result.

17(Original). The method of claim 16, wherein step (c) further includes the steps of:

- c1) comparing a search convergence depth to an access convergence depth;
- c2) when the search convergence depth is equal to the access convergence depth, performing an intersection between the query search result and the access search result to form an intersection result;
- c3) when the access rule is a show command, the allowed search result is the intersection result.

18(Original). The method of claim 17, further including the step of:

- c4) when the search convergence depth is greater than the access convergence depth, converging the query search result to the access convergence depth to form a converged query search result;
- c5) performing an intersection between the convergence search result and the access search result to form the intersection result.

19(Original). The method of claim 16, wherein step (a) further includes the steps of:

- a1) determining a user's organization;
- a2) retrieving an access control rule based on the user's organization.

20(Original). The method of claim 16, wherein step (a) further includes the step of:

- (a1) flattening the extensible markup language file.

21(Previously Presented). A method of control in an extensible markup language file, comprising the steps of:

- a) predefining a pattern as a trigger
- b) searching for the pattern using a sliding window search; and
- c) when a pattern is found, altering the extensible markup language file or creating an extensible markup language document.

22(Original). The method of claim 21, wherein step (a) includes predefining a command as a trigger.

23(Original). The method of claim 21, wherein step (a) includes predefining a tag as a trigger.

24(Original). The method of claim 21, wherein step (a) includes predefining a data as a trigger.

25(Original). The method of claim 21, wherein step (c) includes performing an access control search.

26(Original). The method of claim 21, wherein step (c) includes deleting a portion of the extensible markup language file.

27(Original). The method of claim 21, wherein step (c) includes performing a translation of a portion of the extensible markup language file.